Review

1. A search function is log n because it when searching a binary search tree your either going left and right consistently. By doing this you are decreasing your search by half every time.
2. You put the biggest number at the root of a heap sort when your using the same array in the heap sort when sorting the array from biggest to smallest. So when you pop the biggest value from the top you will put that value and the end of the array and replace the root with the next biggest value.
3. A quick sort will take a pivot from the array or linked list and will use that pivot to put values less than it on one side and values greater than it on the other. It does this by using two pointer or indexes depending if it’s a linked list or an array. When the two pointers or indexes meet it will take that value and swap it with the pivot. The function will then split the linked list or array into the two sides, the values greater than the pivot and the values less than the pivot and repeat the process until one value is left.
4. A merge sort is nlogn because it takes logn runtime to break the array or linked list down to one value each and then n runtime to merge it back together.
5. I would use a hash table to store the phone numbers.
6. It is logn because every time you go left and right your lessening your search by half consistently.
7. No you cannot because linked lists are located in different sections of memory while arrays are sequential. The only way to actually get somewhere close to a binary search tree would be to make an iterator class with an overloaded [] bracket but that to would give you an nlogn runtime and not a logn runtime.